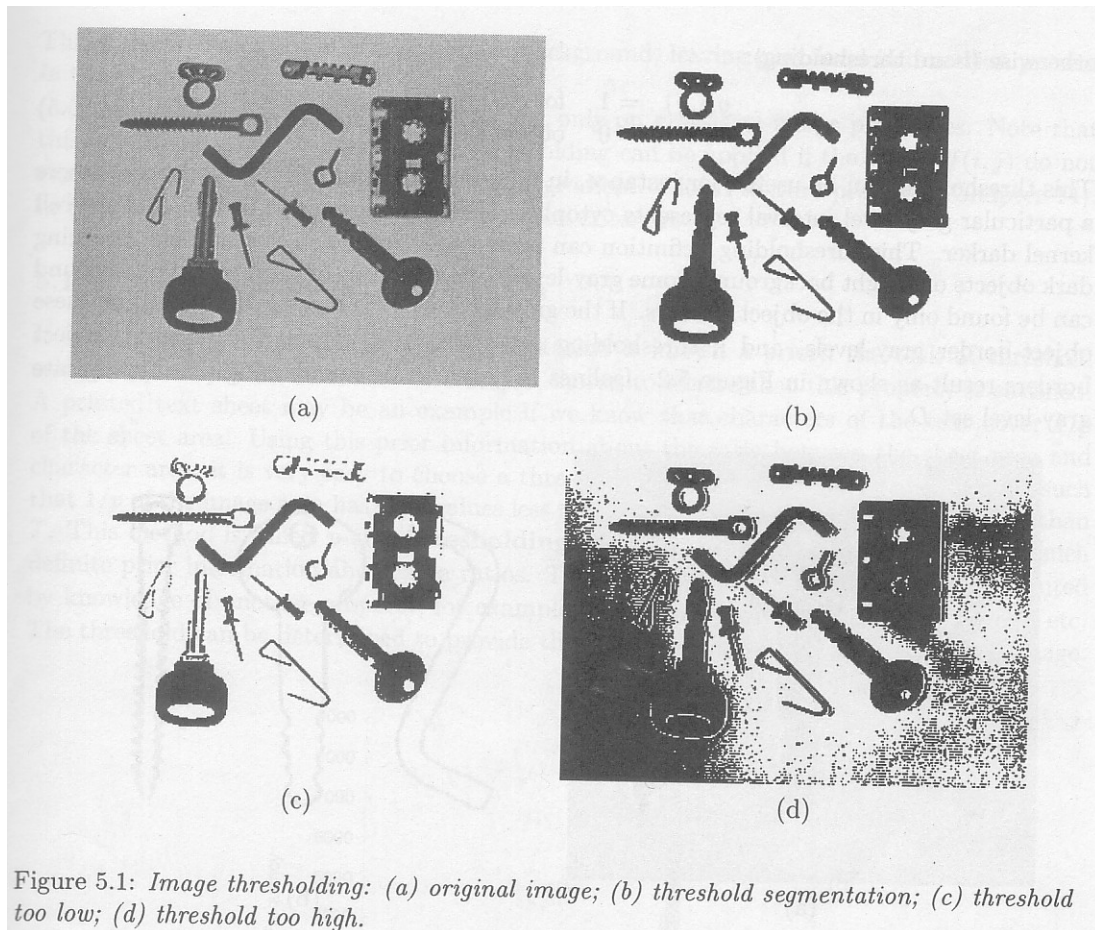


SEGMENTÁCIA OBRAZU

- Hlavným cieľom segmentácie je rozdeliť obraz na časti, ktoré majú silnú koreláciu s objektami alebo oblasťami reálneho sveta zobrazenými v obraze.
- Úplná segmentácia (rozdelenie obrazu na disjunktné oblasti, ktoré zodpovedajú objektom) a čiastočná segmentácia (kde čiastočne zodpovedajú)
- Pri čiastočnej segmentácii je cieľom rozdeliť obraz na časti, ktoré sú homogénne z hľadiska vybranej vlastnosti, napr. jas, farby, odrazivosti, textúry apod.
- Pri segmentácii šedoúrovňových obrazov využívame dve základné vlastnosti týchto obrazov: diskontinuitu a podobnosť.
- Segmentačné metódy možno rozdeliť na tri skupiny: **prahovanie**, segmentácia **založená na hranách** (diskontinuita), segmentácia **založená na oblastiach** (podobnosť).
- Každá oblasť môže byť reprezentovaná uzavretou hranicou a každá uzavretá hranica popisuje oblasť.
- Nejednoznačnosť obrazových dát je hlavným segmentačným problémom, často sprevádzaným informačným šumom.
- Čím viac apriórnej informácie je k dispozícii pri segmentačnom procese, tým lepšie výsledky pri segmentácii možno dosiahnuť.

Prahovanie

- Najjednoduchšia segmentačná technika, je výpočtovo nenáročná a rýchla. Na segmentovanie objektov od pozadia sa používa jasová konštanta, ktorá sa nazýva **prah**.
- Prahovanie je transformácia, ktorá zobrazuje vstupný obraz $f(i, j)$ na výstupný obraz $g(i, j)$ nasledovne:
 - o $g(i, j) = 1$ ak $f(i, j) \geq T$
 $= 0$ ak $f(i, j) < T$
- Prahovanie s jedným prahom možno aplikovať na celý obraz (vtedy ide o globálne prahovanie), alebo sa prah môže meniť v jednotlivých častiach obrazu (lokálne prahovanie). Iba za veľmi špeciálnych okolností môže byť prahovanie úspešné s jedným prahom v celom obraze.
 - o globálne prahovanie $T = T(f)$
 - o lokálne prahovanie $T = T(f, f_c)$, kde f_c je časť obrazu, v ktorej sa určuje prah.



- Existuje viacero modifikácií základného prahovania: intervalové prahovanie, poloprahovanie, prahovanie s viacerými prahmi.

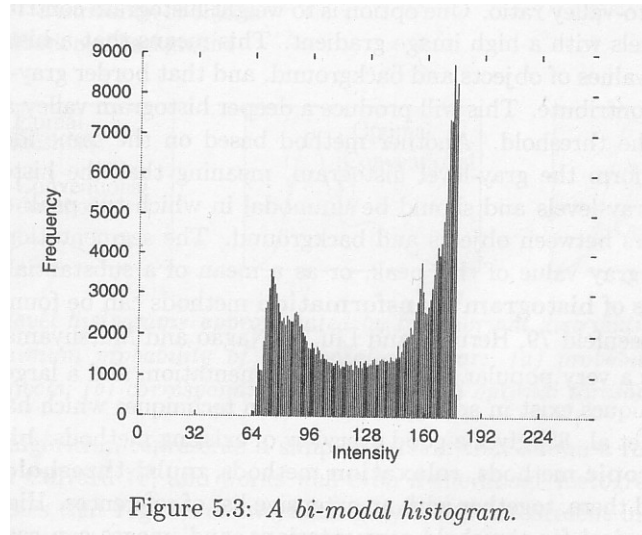
- o intervalové
$$g(i, j) = \begin{cases} 1 & \text{ak } f(i, j) \in D \\ 0 & \text{inak.} \end{cases}$$

- o poloprahovanie
$$g(i, j) = \begin{cases} f(i, j) & \text{ak } f(i, j) \geq T \\ 0 & \text{ak } f(i, j) < T \end{cases}$$

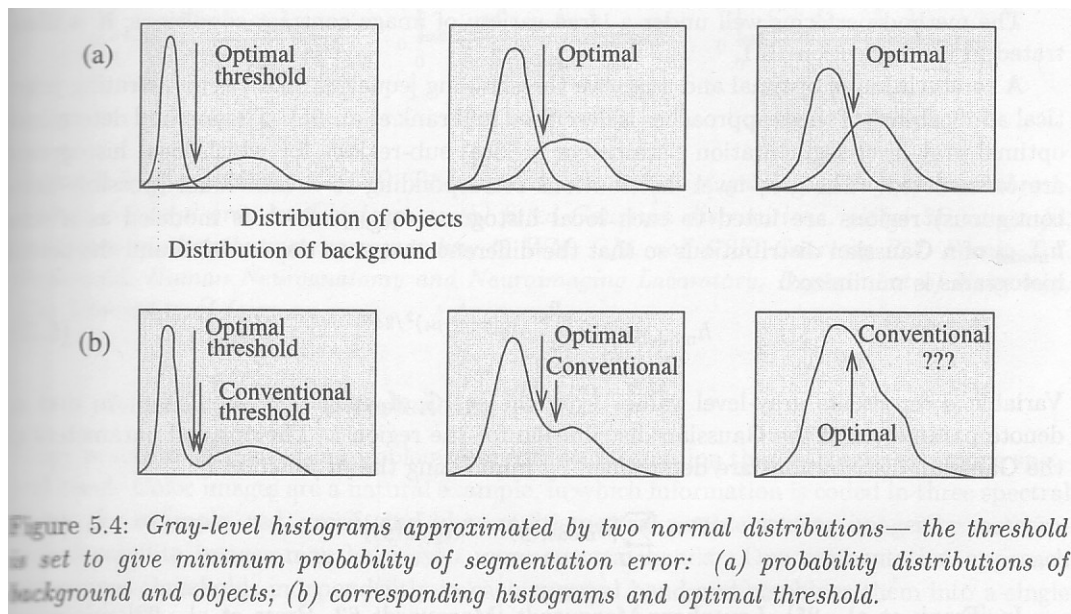
- o s viacerými prahmi
$$g(i, j) = \begin{cases} 1 & \text{ak } f(i, j) \in D_1 \\ 2 & \text{ak } f(i, j) \in D_2 \\ 3 & \text{ak } f(i, j) \in D_3 \\ \dots & \dots \\ n & \text{ak } f(i, j) \in D_n \\ 0 & \text{inak.} \end{cases}$$

- Metódy na určenie prahu sa snažia určiť prah automaticky. Na určenie prahu sa používajú *metóda p-podielu*, *metóda analýzy tvaru histogramu* a *optimálne prahovanie*.

- Metóda p-podielu. Ak sa vie o nejakej vlastnosti, ktorú má spĺňať obraz po segmentácii, tak je úloha prahovania zjednodušená, pretože prah sa vyberá tak, aby obraz túto vlastnosť spĺňal. Konkrétne ak vieme, že objekt pokrýva $1/p$ obrazovej plochy, tak vyberáme prah T tak, by objekt (napr. písmeno) pokrývalo takúto plochu a pozadie zvyšok.
- v bimodálnych histogramoch sa vyberá prah ako minimum medzi dvoma najvyššími lokálnymi maximami

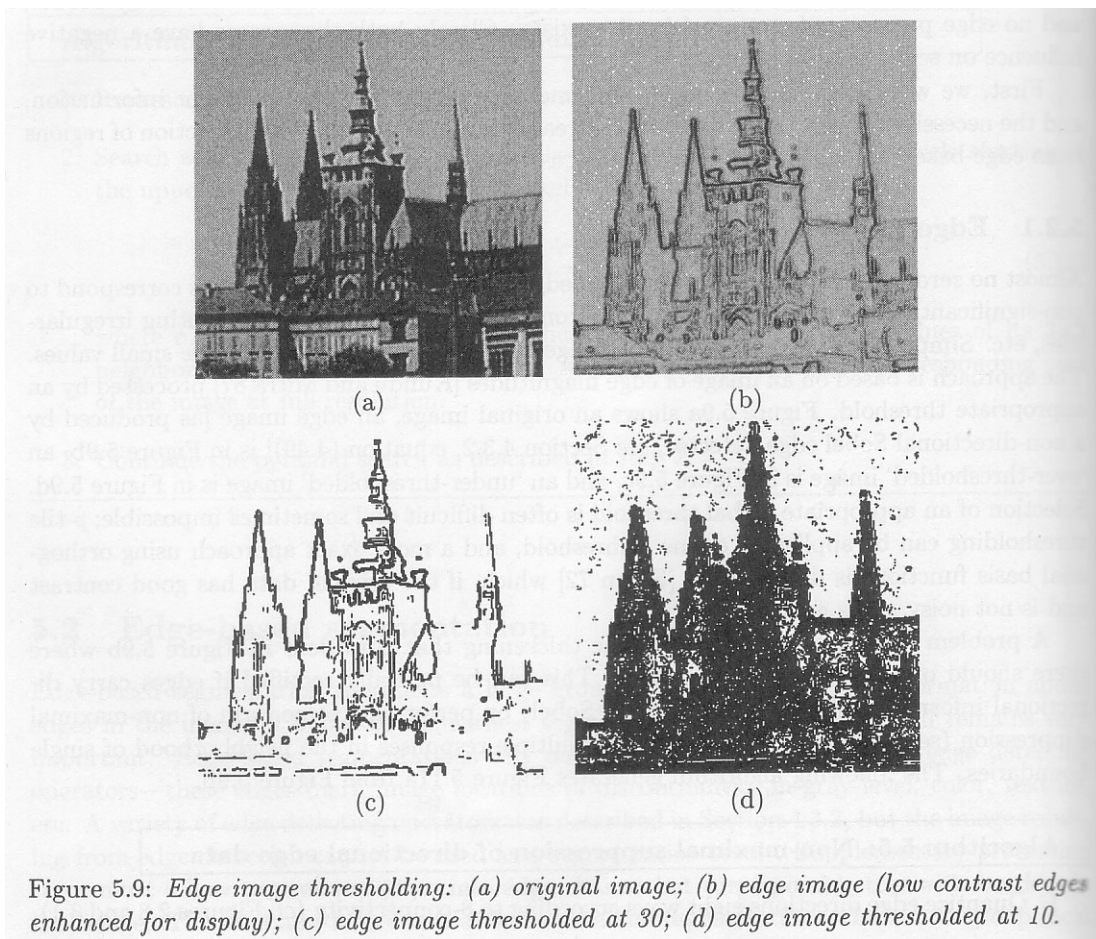


- optimálne prahovanie určuje prah ako najbližšiu šedú úroveň zodpovedajúcu minimu pravdepodobnosti medzi maximami dvoch alebo viacerých normálnych rozdelení. Takéto prahovanie vedie k minimálnej chybe pri segmentácii.



Segmentácia založená na hranách

- Je založená na lokálnych hranách, ktoré boli detekované hranovými operátormi. Lokálne hrany zodpovedajú ostrému prechodu v úrovni šedej, vo farbe, v textúre apod.
- Najväčšími problémami segmentácie založenej na hranách, ktoré sú spôsobené obrazovým šumom alebo nevhodnou informáciou v obraze, je prítomnosť lokálnych hrán v miestach, kde neexistuje globálna hranica a naopak neprítomnosť lokálnej hrany tam, kde globálna hranica existuje.
- **1. Prahovanie obrazu hrán** je založené na vytvorení lokálnych hrán, ktoré sa prahujú vhodným prahom (predpokladá sa, že silnejšie lokálne hrany sú časťou globálnej hranice)



- **2. Pri relaxácii (optimalizácii) hrán** sa vlastnosti hrany posudzujú v kontexte okolitých hrán. Ak existujú dostatočné príznaky existencie hranice, lokálna hrana sa posilňuje a naopak. Napr. slabá hrana medzi dvoma silnými hranami, je veľmi pravdepodobné, že táto slabá hrana bude súčasťou globálnej hranice. Ak sa použije globálna relaxácia (optimalizácia), vytvoria sa súvislé hrany.

- o pracuje sa s pojmom dôvery hrany, iteratívne sa posudzuje dôvera každej hrany, až kým nekonverguje k 0 alebo k 1.
- o 0-0 – izolovaná hrana, negatívny vplyv na dôveru hrany
 0-1 – neurčitá, slabo pozitívna, neutrálny vplyv na dôveru hrany
 0-2 , 0-3 – neperspektívna hrana, negatívny vplyv na dôveru
 1-1 – pokračovanie, silný pozitívny vplyv na dôveru hrany
 1-2 , 1-3 – pokračovanie k priesečníku hranice, stredný pozitívny vplyv
 2-2, 2-3, 3-3 – most medzi hranicami- nie je potrebný pre segmentáciu, žiadny vplyv na dôveru hrany

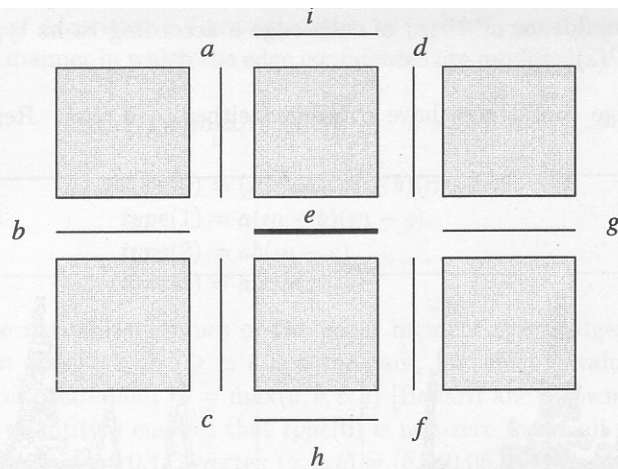


Figure 5.12: Crack edges surrounding central edge e .

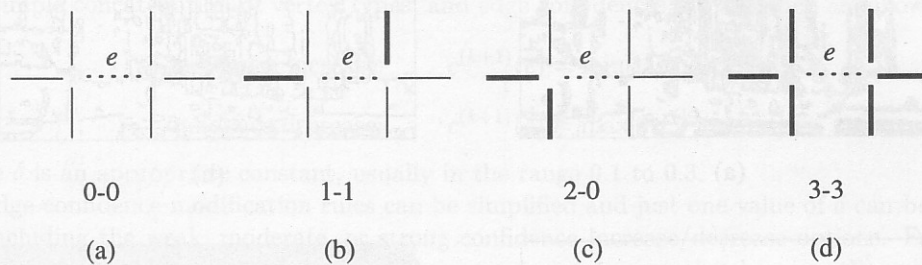


Figure 5.13: Edge patterns and corresponding edge types.

- **2-2, 2-3, 3-3** bridge between borders—not necessary for segmentation, no influence on edge confidence

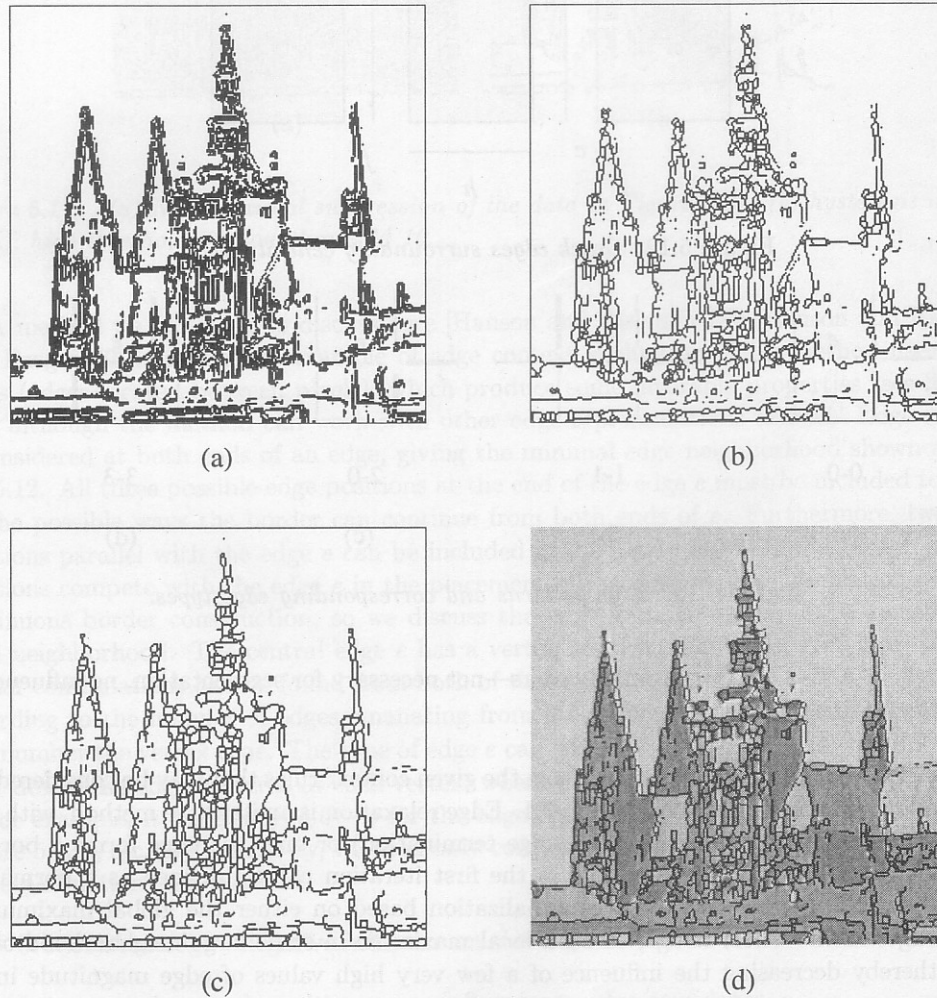


Figure 5.14: *Edge relaxation, see Figure 2.3a for original: (a) resulting borders after 10 iterations; (b) borders after thinning; (c) borders after 100 iterations, thinned; (d) borders after 100 iterations overlaid over original.*

- Ak je definované kritérium optimality, potom na určenie globálne optimálnych hraníc možno použiť **(heuristické) grafové hľadanie** alebo **dynamické programovanie**. Detekcia hranice založená na prehľadávaní grafu je veľmi účinný nástroj na segmentáciu obrazu.

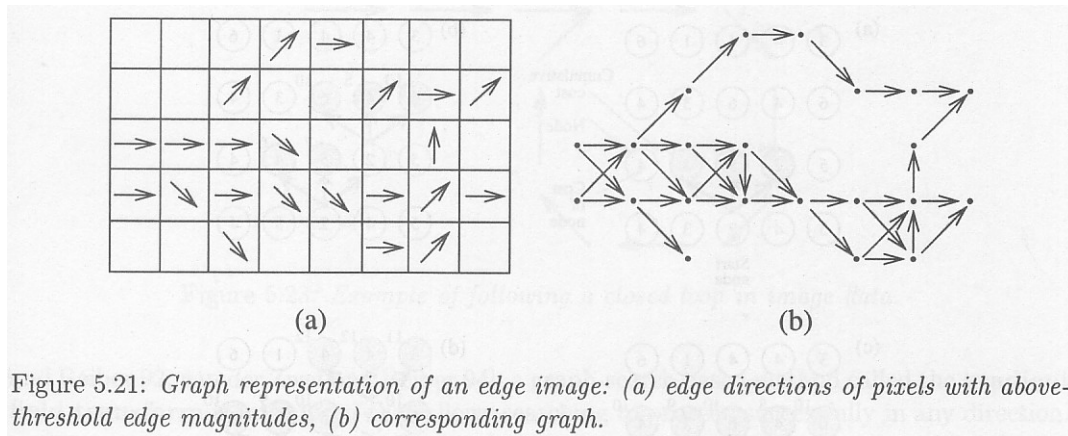
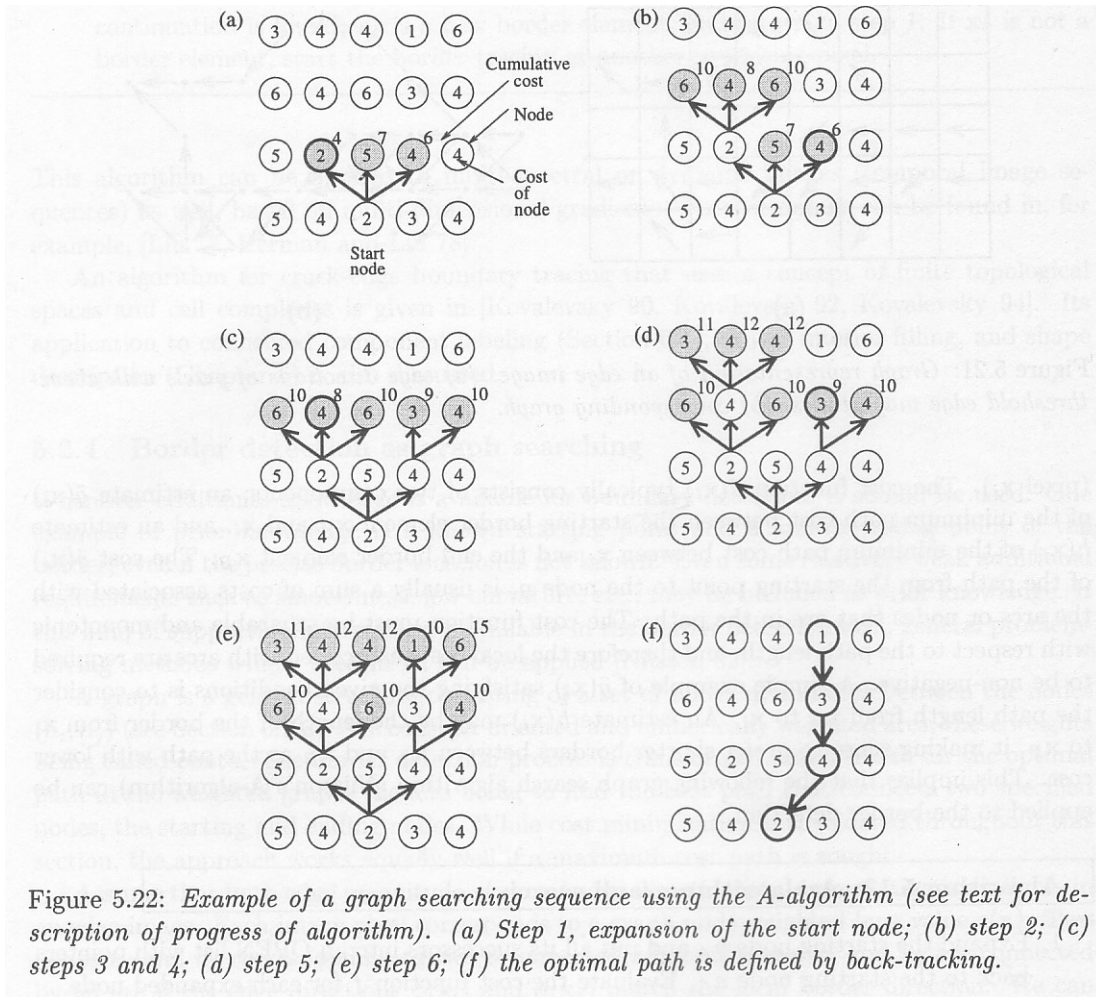


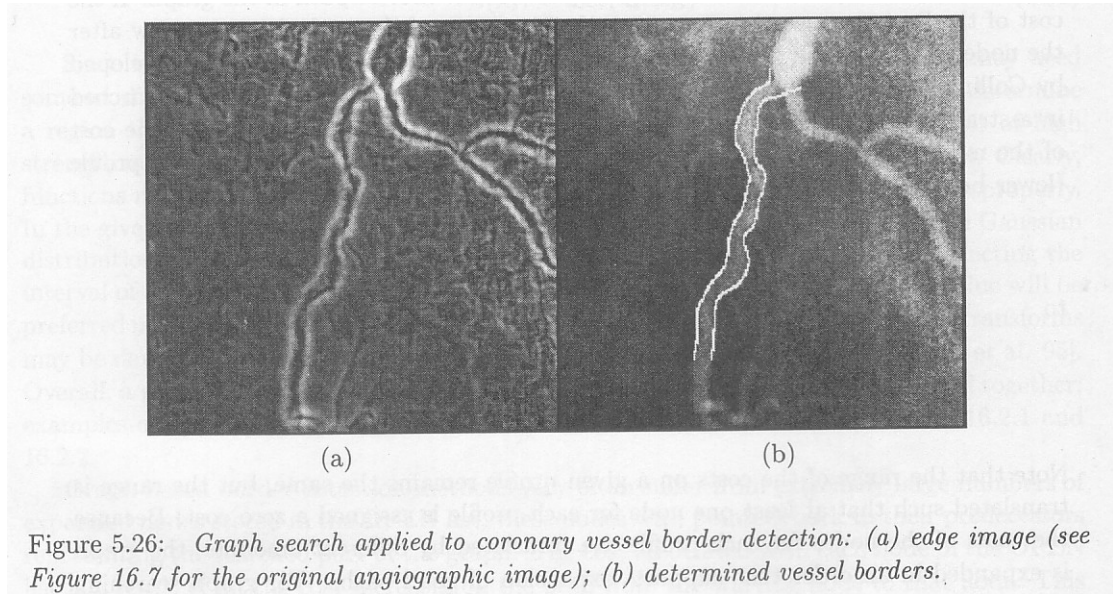
Figure 5.21: Graph representation of an edge image: (a) edge directions of pixels with above-threshold edge magnitudes, (b) corresponding graph.

- **3. Určovanie hranice ako heuristické hľadanie.** Proces určovania hranice sa transformuje na hľadanie optimálnej cesty v ohodnotenom grafe. Ohodnotenia (ceny) sú spojené s každým vrcholom a vyjadrujú šancu, že hranica povedie cez tento vrchol (pixel). Cieľom je nájsť optimálnu hranicu (vzhľadom na určité optimalizačné kritérium), ktorá spája dva špecifikované obrazové body alebo množiny obrazových bodov, ktoré reprezentujú začiatok a koniec hranice.
- **Definícia ceny** (ohodnocujúce funkcie) sú kľúčom k úspešnej detekcii hranice. Zložitosť výpočtu ceny vrcholu môže byť taká, že je to prevrátená hodnota veľkosti lokálnej hrany v danom bode až po apriórnu znalosť o hľadanej hranici, cieľi segmentácie, obrazových dátach apod.
- Prehľadávanie grafu používa Nilssonov algoritmus A^* , ktorý zaručuje optimálny výsledok. Požíva ohodnocovaciu funkciu v tvare $f(n) = g(n) + h(n)$, kde g zložka reprezentuje tú časť grafu, ktorá už bola prehľadaná a vyjadruje cenu cesty, ktorá vedie od počiatočného vrcholu do vrcholu n a h zložka je heuristická zložka, ktorá odhaduje cenu cesty, ktorá vedie z vrcholu n do niektorého z koncových vrcholov. Algoritmus vyberá na ďalšie expandovanie taký vrchol, ktorý má minimálne $f(n)$, pričom berie do úvahy všetky *otvorené* vrcholy, t.j. tie, ktoré ešte neboli expandované.



- Heuristické prehľadávanie môže výrazne urýchliť prehľadávanie grafu, ale heuristika musí spĺňať určité vlastnosti, ak má byť zaručená optimalita. Heuristika musí byť prípustná, to znamená, že $0 \leq h \leq h^*$, kde h^* reprezentuje optimálnu heuristiku. Kvalita hľadania je tým lepšia, čím viac sa h blíži ku h^* . Ak $h = 0$, tak sa heuristické hľadanie zmení na hľadanie do šírky. Ak je heuristika prípustná, algoritmus nájde optimálne riešenie, ak nejaké riešenie existuje.
- Ako efektívne zvolit' cenovú funkciu, ktorá ohodnocuje hrany v grafe:
 - o sila hrany tvoriacej hranicu (vychádza z predpokladu, že čím silnejšie sú hrany, ktoré tvoria hranicu, tým väčšia pravdepodobnosť, že tadiaľ prechádza hranica):
 - $\text{cena hrany} = (\max s(x_k)) - s(x_i)$
 - o zakrivenie hranice, používa sa vtedy, ak sa preferujú malé zakrivenia hranice
 - $\text{zakrivenie} = \text{diff}[\phi(x_i) - \phi(x_j)]$, kde diff je funkcia ohodnocujúca rozdiel v smeroch dvoch nasledujúcich prvkov hranice
 - o blízkosť ku známej polohe hranice, ak sa vie približné umiestnenie hranice, potom je treba podporovať hrany, ktoré ležia bližšie k predpokladanej hranici.

- vzdialenosť = $dist(x_i, \text{približná_hranica})$
- blízkosť ku koncovému bodu, ak je hranica priamočiara, je prirodzené podporovať body, ktoré ležia bližšie ku koncovému bodu
 - heuristika $h(x_i) = dist(x_i, x_B)$, kde x_B je koncový bod hranice



- Problémy s touto metódou sú tie, že obsahuje veľké množstvo vrcholov, ktoré sú otvorené a ktoré treba rozvíjať, aj keď je málo pravdepodobné, že cez ne nepovedie hľadanie cesta, niekedy sa používajú modifikácie algoritmu: redukcia počtu otvorených vrcholov, počítanie ceny cesty len ako súčet silnejších hrán, vylúčenie ciest, ktoré presiahnu stanovenú cenu cesty.
- **4. Dynamické programovanie** je založené na princípe optimality a reprezentuje efektívnu cestu simultánneho hľadania optimálnych ciest z viacerých začiatkových a koncových bodov. Hľadá optimálnu funkciu, v ktorých nie všetky premenné sú navzájom závislé.
- Keď sa používa algoritmus A* na prehľadávanie grafu, nie je nutné vytvárať celý graf, pretože ohodnotenie expandovaných vrcholov sa počíta z doteraz generovaného grafu, kým v dynamickom programovaní musí byť vytvorený celý graf.

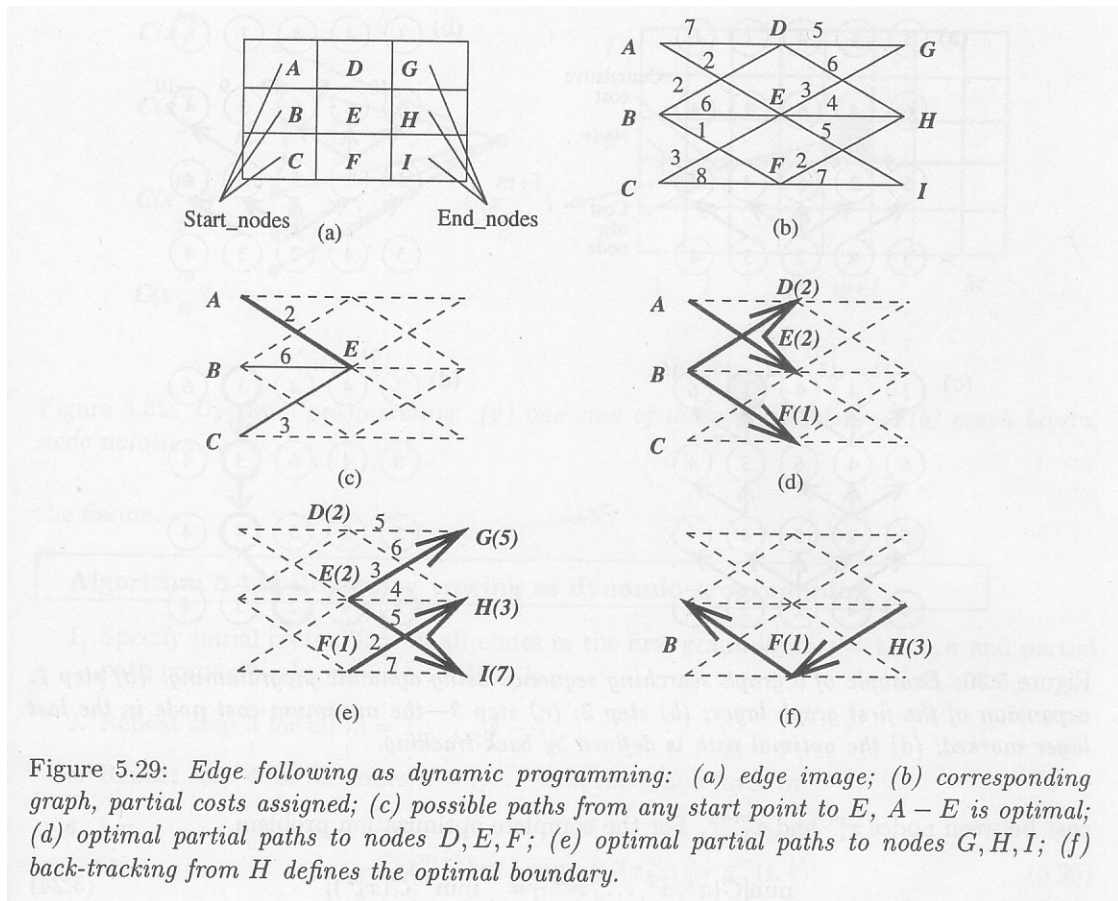


Figure 5.29: Edge following as dynamic programming: (a) edge image; (b) corresponding graph, partial costs assigned; (c) possible paths from any start point to E, A – E is optimal; (d) optimal partial paths to nodes D, E, F; (e) optimal partial paths to nodes G, H, I; (f) back-tracking from H defines the optimal boundary.

- Hlavná myšlienka princípu optimality je: Akákoľvek bola cesta z počiatočného vrcholu do E, tak existuje optimálna cesta z E do koncového vrcholu. Inými slovami, ak existuje optimálna cesta z počiatočného do koncového vrcholu, ktorá ide cez E, tak obidve podcesty z počiatočného vrcholu do E a z E do koncového vrcholu musia byť tiež optimálne.
- keď má graf viacero vrstiev, proces sa opakuje, kým sa nedosiahne jeden z koncových vrcholov. Každé opakovanie pozostáva z jednoduchej optimalizácie, ktorá je ilustrovaná na nasledujúcom obrázku

o $C(x_k^{m+1}) = \min_i [C(x_i^m) + g^m(i, k)]$, kde $C(x_k^{m+1})$ je nová hodnota priradená vrcholu

x_k^{m+1} , a $g^m(i, k)$ je čiastočná cesta medzi vrcholmi x_i^m a x_k^{m+1} . Pre celkový optimalizačný problém je

- $\min[C(x^1, x^2, \dots, x^M)] = \min_{k=1, \dots, n} [C(x_k^M)]$ kde $C(x_k^M)$ sú koncové vrcholy, M počet úrovní medzi počiatočnými a koncovými vrcholmi a $[C(x^1, x^2, \dots, x^M)]$ označuje cenu cesty medzi prvou a poslednou (M-tou) úrovňou.

- Ak budeme uvažovať 8-susednosť, n vrcholov v každej úrovni, tak v každej úrovni treba počítať $3n$ kombinácií cien a celkove $3n(M-1) + n$ výpočtov, čo v porovnaní s úplným preberaním, ktoré vyžaduje $n(3^{M-1})$ výpočtov je značná úspora.

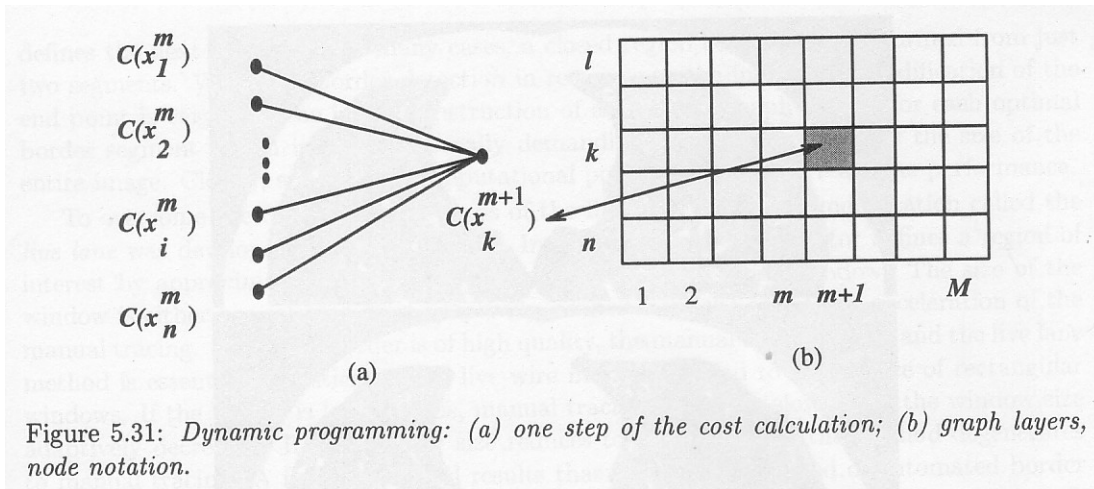


Figure 5.31: Dynamic programming: (a) one step of the cost calculation; (b) graph layers, node notation.

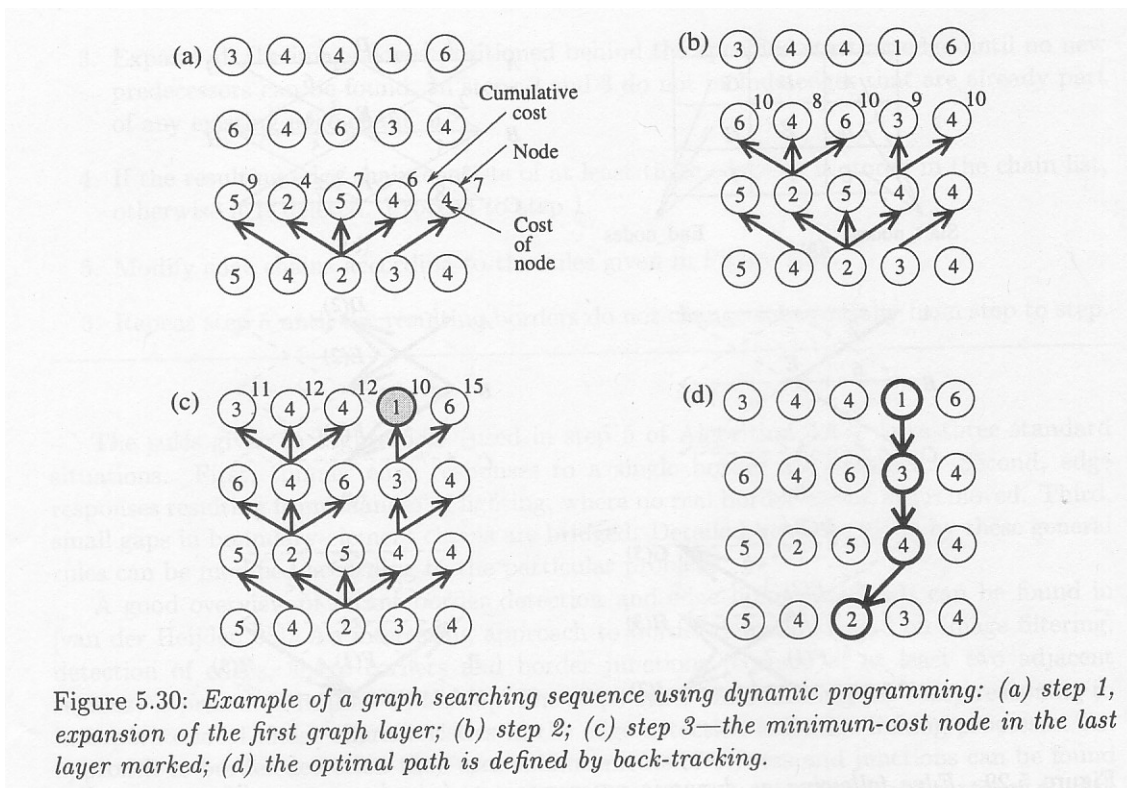
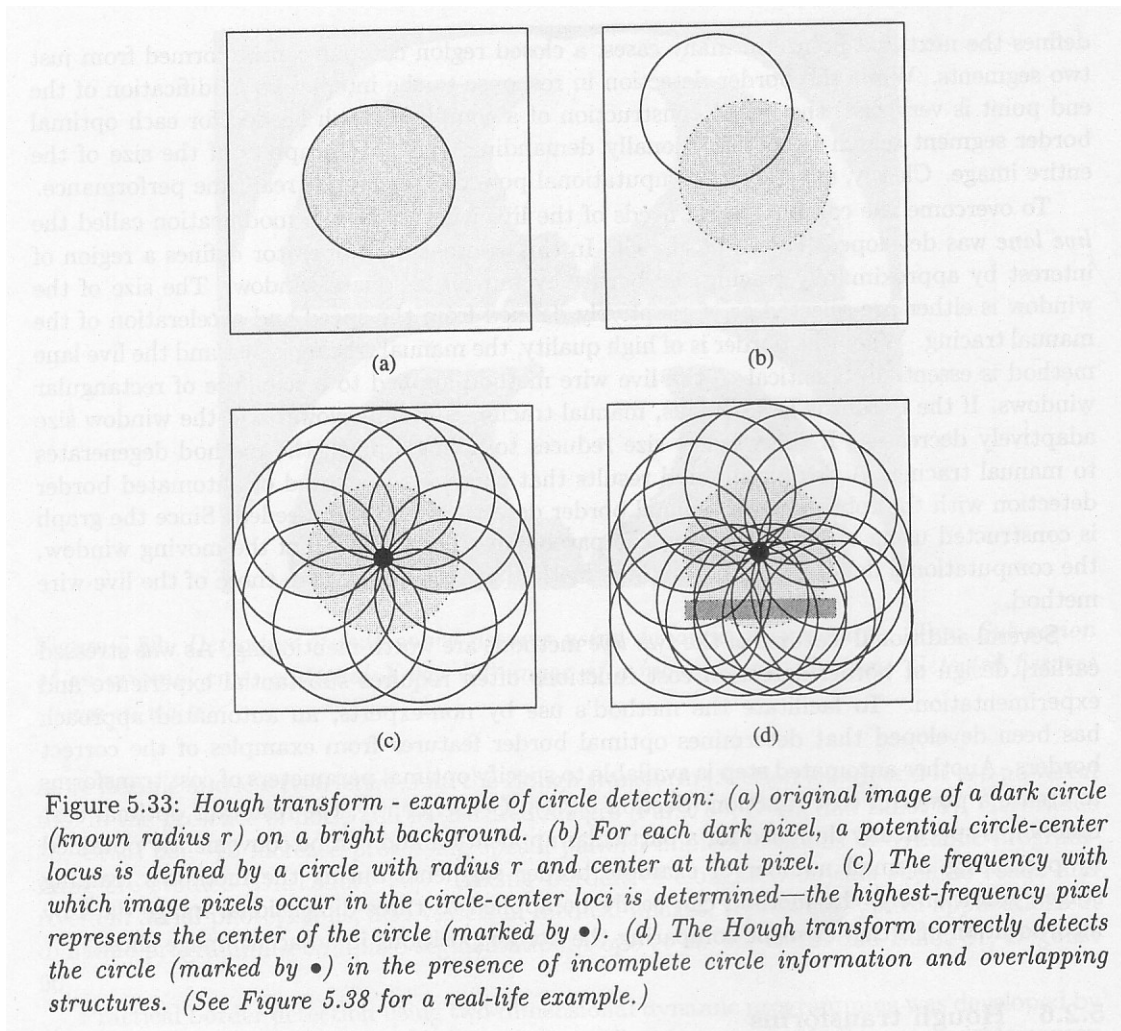


Figure 5.30: Example of a graph searching sequence using dynamic programming: (a) step 1, expansion of the first graph layer; (b) step 2; (c) step 3—the minimum-cost node in the last layer marked; (d) the optimal path is defined by back-tracking.

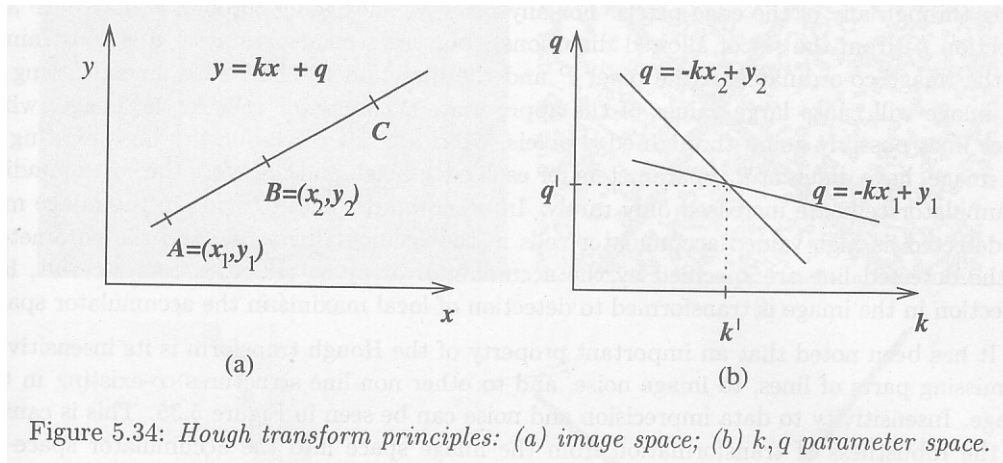
- Ak počítanie lokálnej ohodnocovacej funkcie je výpočtovo nenáročné, dynamické programovanie môže byť výpočtovo menej náročnou alternatívou. Avšak otázka, ktorý z dvoch prehľadávacích prístupov (algoritmus A*, dynamické programovanie) bude

efektívnejší pre konkrétny problém závisí na ohodnocovacej funkcii a kvalite heuristiky pre algoritmus A*.

- **5. Houghova transformácia** je segmentačná technika použiteľná vtedy, keď treba detekovať objekty so známym tvarom hranice. Houghova transformácia môže detekovať rovné čiary aj krivky (hranice objektu), ak sú známe ich analytické vyjadrenia. Je robustná pri identifikácii zakrytých a zašumených objektov
- Ako príklad uveďme detekciu kruhu, ktorý pozostáva z tmavých bodov.



- Pôvodne bola Houghova transformácia navrhnutá na detekciu rovných čiar.



- Máme známe body, ktoré ležia na priamke a máme jej parametrické vyjadrenie. Tieto známe body dosadíme a hľadáme také parametre k' a q' , ktoré najlepšie zodpovedajú týmto známym bodom. Bod (k', q') v parametrickom priestore určuje výslednú priamku.
- Praktické použitie: na obraz aplikujeme hranový operátor. Počet priamok prechádzajúcich cez tieto lokálne hrany je prakticky neobmedzený, ale v praxi sa uvažuje len konečný počet možných smerov, ktorý reprezentuje diskretizáciu parametra k a podobne parameter q sa môže vzorkovať na obmedzený počet hodnôt. Takže namiesto spojitého parametrického priestoru vznikne obdĺžniková štruktúra buniek, ktorá sa nazýva akumulčné pole A s bunkami $A(k, q)$. Pre každý pixel hrany, sa určia parametre k, q , ktoré možné priamky prechádzajú cez tento pixel. Tieto zväčšia hodnotu príslušného bodu $A(k, q)$. Na konci zostanú bunky, ktoré majú vysoké akumulované hodnoty, kým iné kombinácie majú nízke hodnoty a nerepresentujú skutočné priamky.

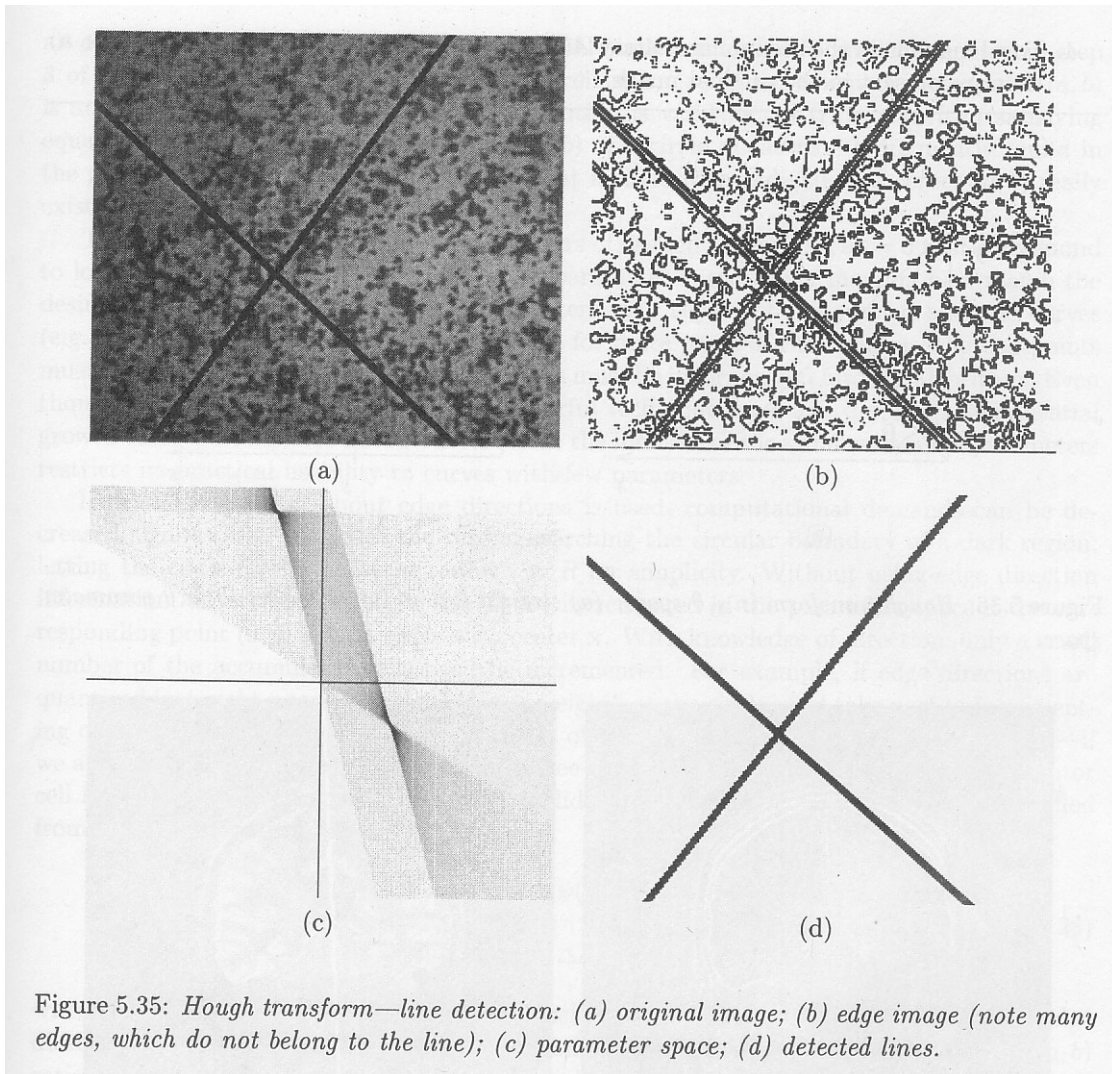


Figure 5.35: *Hough transform—line detection: (a) original image; (b) edge image (note many edges, which do not belong to the line); (c) parameter space; (d) detected lines.*

- Namiesto reprezentácie $y = kx + q$, ktorá má problémy pre zvislé línie ($k \rightarrow \infty$) sa používajú polárne súradnice, kde priamka je reprezentovaná ako $s = x \cos \phi + y \sin \phi$ a nemá tieto problémy.
- Diskretizácia parametrického priestoru nie je až taká jednoduchá a aj nájdenie lokálneho maxima akumuláčného poľa nemusí byť také jednoduché.

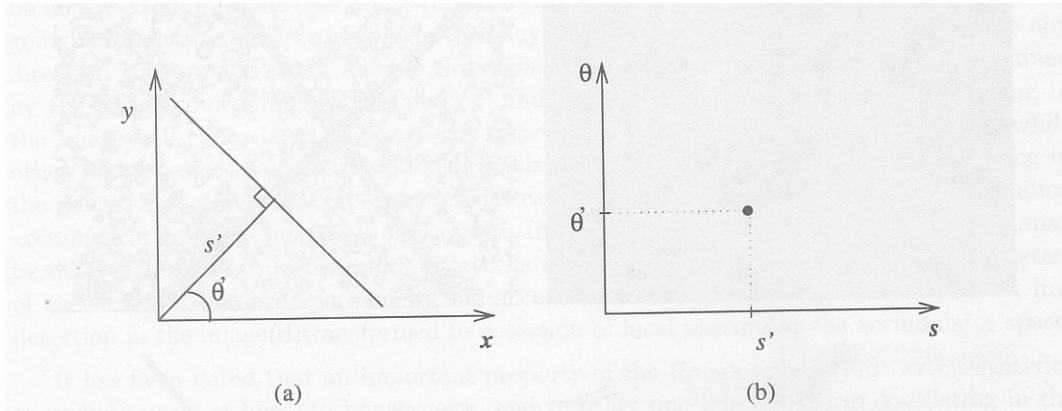


Figure 5.36: *Hough transform in s, θ space: (a) straight line in image space; (b) s, θ parameter space*

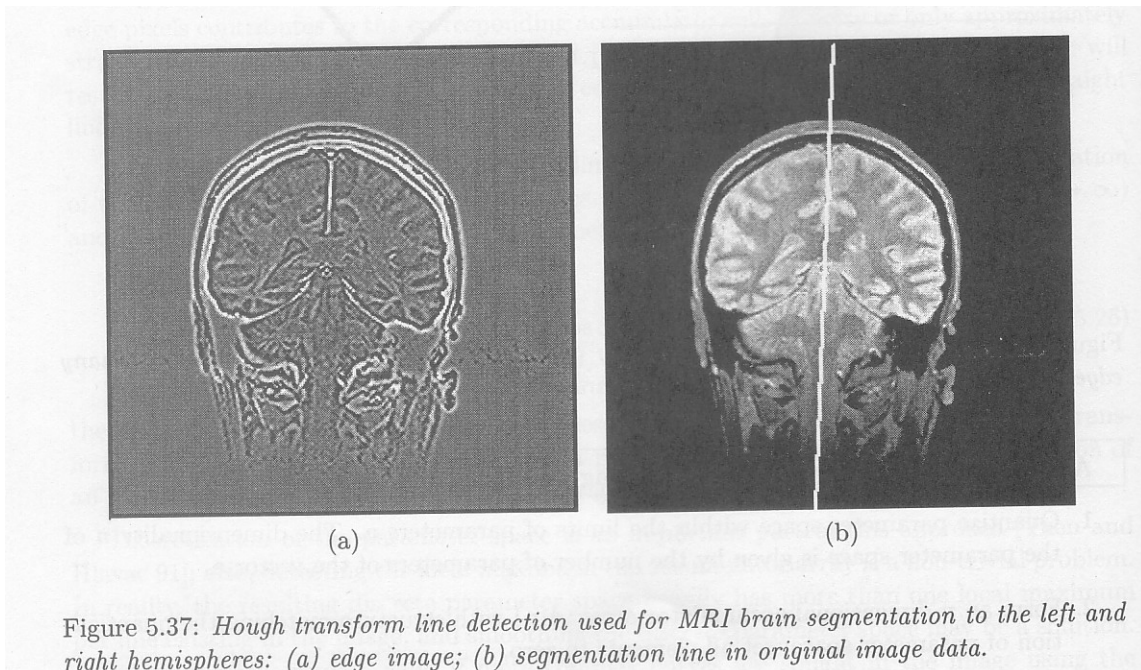
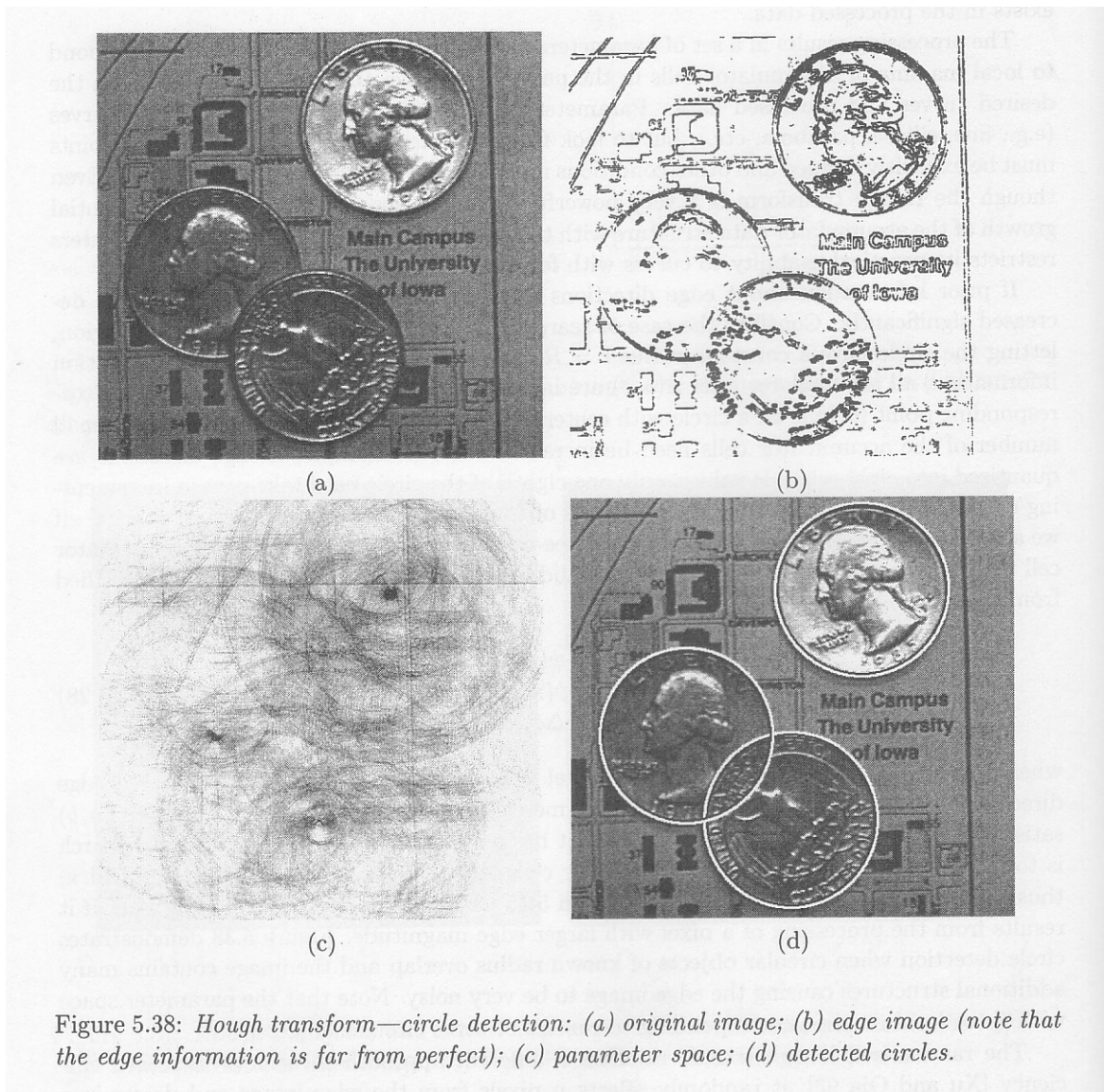


Figure 5.37: *Hough transform line detection used for MRI brain segmentation to the left and right hemispheres: (a) edge image; (b) segmentation line in original image data.*

- Rovnako sa dá postupovať pri hľadaní kriviek. Napr. pri kružnici budeme mať trojrozmerný parametrický priestor (a, b, r), takže aj akumulčné pole bude trojrozmerné. Výsledky môžu byť veľmi dobré, ako ukazuje nasledujúci obrázok.



- Zovšeobecnenú Houghovu transformáciu možno použiť vtedy, ak analytické vyjadrenia hľadaných tvarov nie sú k dispozícii; popis parametrických kriviek (hraníc oblastí) je založený na vzorových situáciách a je určený stavom učenia.

Segmentácia založená na oblastiach

- Segmentácia založená na oblastiach musí spĺňať nasledovné kritérium úplnej segmentácie:

$$R = \bigcup_{i=1}^S R_i \quad R_i \cap R_j = 0 \quad i \neq j$$

a podmienky maximálnej homogenity:

$$H(R_i) = \text{TRUE}, \text{ pre } i = 1, 2, \dots, S$$

$$H(R_i \cap R_j) = \text{FALSE} \text{ pre } i \neq j \text{ a } R_i \text{ je susedné s } R_j$$

- Kritérium homogenity môže byť priemerná šedá úroveň oblasti, farebné vlastnosti, textúrne vlastnosti, tvarové vlastnosti apod.
- Existujú tri základné prístupy k segmentácii založenej na oblastiach: spájanie oblastí, delenie oblastí a delenie-a-spájanie oblastí.
- **1. Spájanie oblastí** začína pri presegmentovanom obraze (rozdeleným na malé oblasti), ktorý spĺňa podmienku, že každá oblasť obrazu je homogénna. Oblasti sa spájajú tak, aby spĺňali druhú podmienku, teda aby boli maximálne za podmienky, že homogenita každej oblasti zostane zachovaná. Jednotlivé metódy sa líšia podľa toho, ako zvolia počiatočnú segmentáciu a kritérium pre spájanie. Výsledná segmentácia závisí obvykle aj na poradí, v akom spájame jednotlivé oblasti.

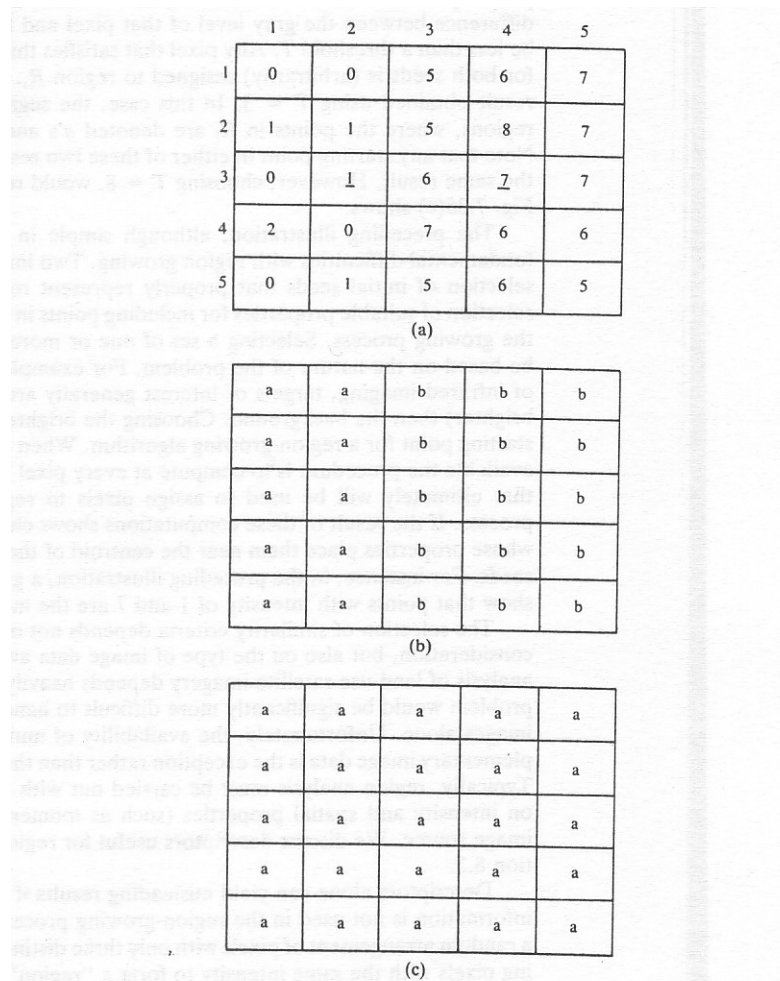
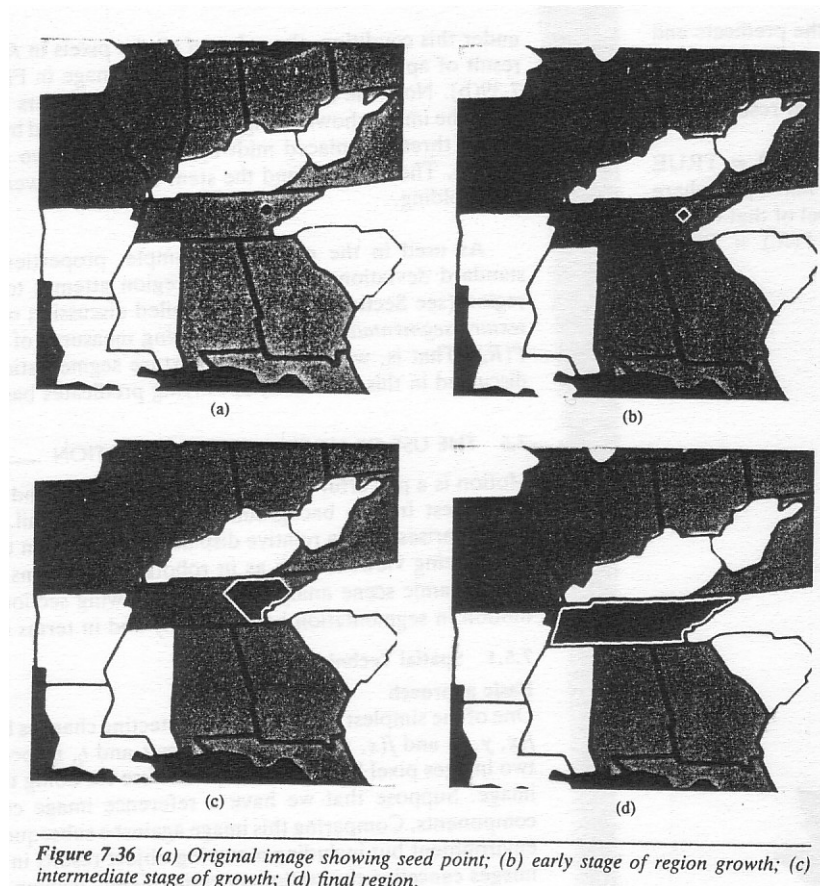
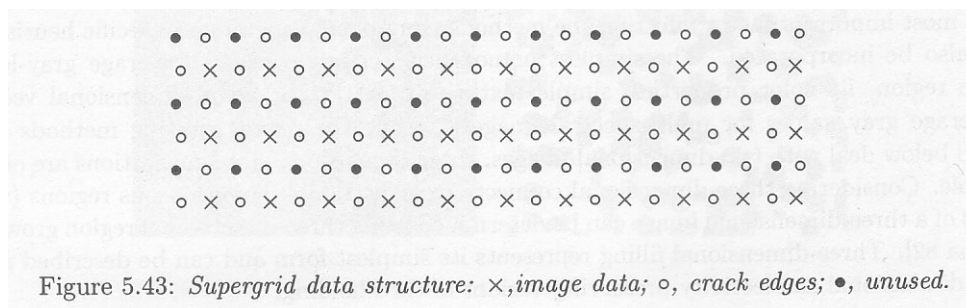


Figure 7.35 Example of region growing using known starting points: (a) original image array; (b) segmentation result using an absolute difference of less than 3 between intensity levels; (c) result using an absolute difference of less than 8.



- Najjednoduchšie metódy začínajú s oblasťami 2 x 2, 4 x 4, 8 x 8. Na spájanie sa používa dátová štruktúra, ktorá sa volá supermriežka a ktorá umožňuje hovoriť aj o priestore medzi pixlami.



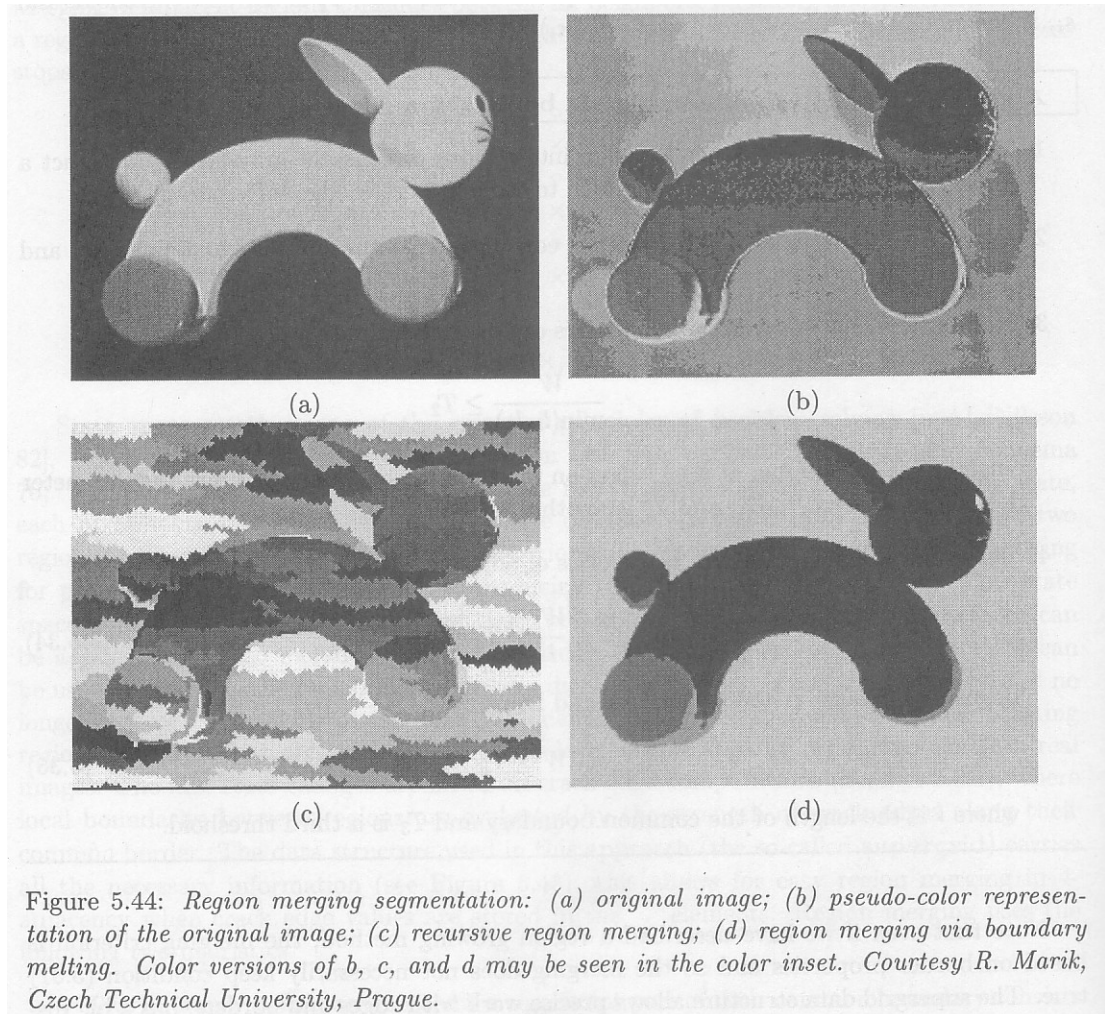
- Na spájanie oblastí sa používajú dve heuristiky:
 - dve susedné oblasti sa spoja, keď výrazná časť ich spoločnej hranice pozostáva zo slabých hrán (napr. počet slabých hrán ku celkovej dĺžke hranice)

- dve susedné oblasti sa spoja, keď výrazná časť ich spoločnej hranice pozostáva zo slabých hrán, ale bez uvažovania o celkovej dĺžke hranice, táto metóda neberie do úvahy vplyv rozdielnej veľkosti oblastí

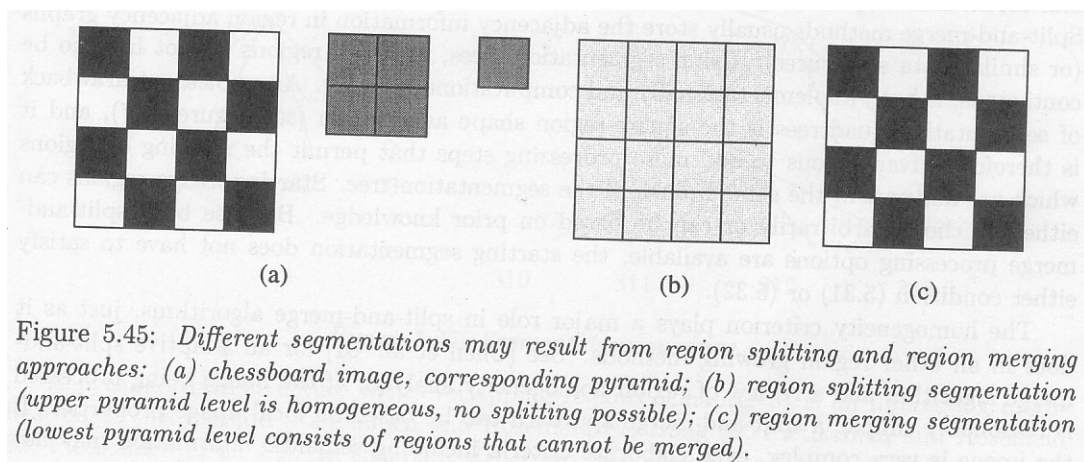
- Významnosť hrany sa určí podľa vzorca
 - $v_{ij} = 0$, ak $s_{ij} < T_1$
 $= 1$ inak,
 - kde hodnota 1 reprezentuje významnú hranu a nula slabú hranu, T_1 je stanovený prah s_{ij} je hodnota hrany medzi pixlami, t.j. $s_{ij} = |f(x_i) - f(x_j)|$.

- Hranica medzi dvoma oblasťami R_i, R_j sa odstraňuje rekurzívne vtedy, ak
 - $\frac{W}{\min(l_i, l_j)} \geq T_2$, kde W je počet slabých hrán v spoločnej hranici, l_i, l_j sú dĺžky priemerov oblastí R_i a R_j a T_2 je ďalší prah.

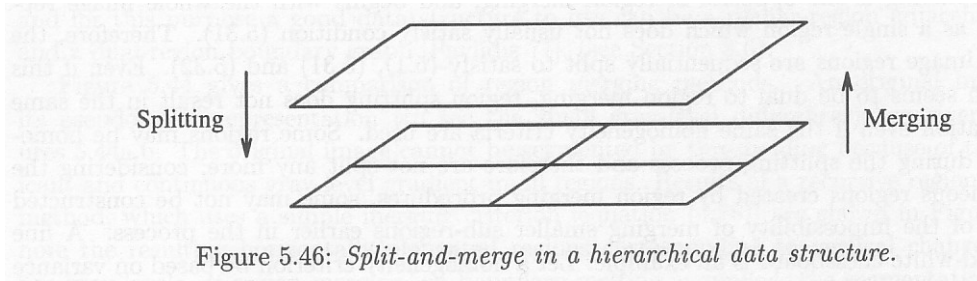
- Hranica medzi dvoma oblasťami R_i, R_j sa odstraňuje rekurzívne vtedy, ak
 - $\frac{W}{l} \geq T_3$, kde W je počet slabých hrán v spoločnej hranici, l je dĺžka spoločnej hranice a T_3 je ďalší prah.



- **2. Delenie oblastí** je opačné ku spájaniu oblastí. Začína pri podsegmentovanom obraze, ktorý nespĺňa podmienku homogenity oblasti. Potom sa existujúce oblasti postupne delia tak, aby spĺňali základné podmienky.



- **3. Delenie-a-spájanie oblastí.** Kombinácia delenia a spájania vedie k metóde, ktorá má výhody oboch prístupov. Techniky delenia-a-spájania obyčajne používajú pyramidové obrazové reprezentácie. Pretože je k dispozícii aj delenie aj spájanie, počiatočný obrázok nemusí spĺňať ani podmienku homogenity oblasti ani maximálnosti homogénnej oblasti.



- Príslušný algoritmus je nasledovný:
 1. Definuj počiatočnú segmentáciu, kritérium homogenity a pyramidálnu dátovú štruktúru
 2. Ak niektorá oblasť R nie je homogénna, rozdeľ ju na 4 podoblasti. Ak ktorékoľvek 4 oblasti s rovnakým rodičom možno spojiť do homogénnej oblasti, spoj ich. Ak už nie je čo deliť a spájať, choď na 3.
 3. Ak ktorékoľvek susedné oblasti R_i a R_j (aj keď sú na rozličných úrovniach pyramidy alebo nemajú spoločného otca) sa dajú spojiť do homogénnej oblasti, spoj ich.
 4. Ak je nevyhnutné odstrániť malé oblasti, spoj ich s najpodobnejšími susednými oblasťami.

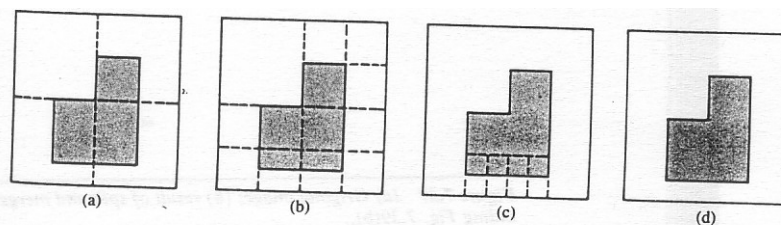
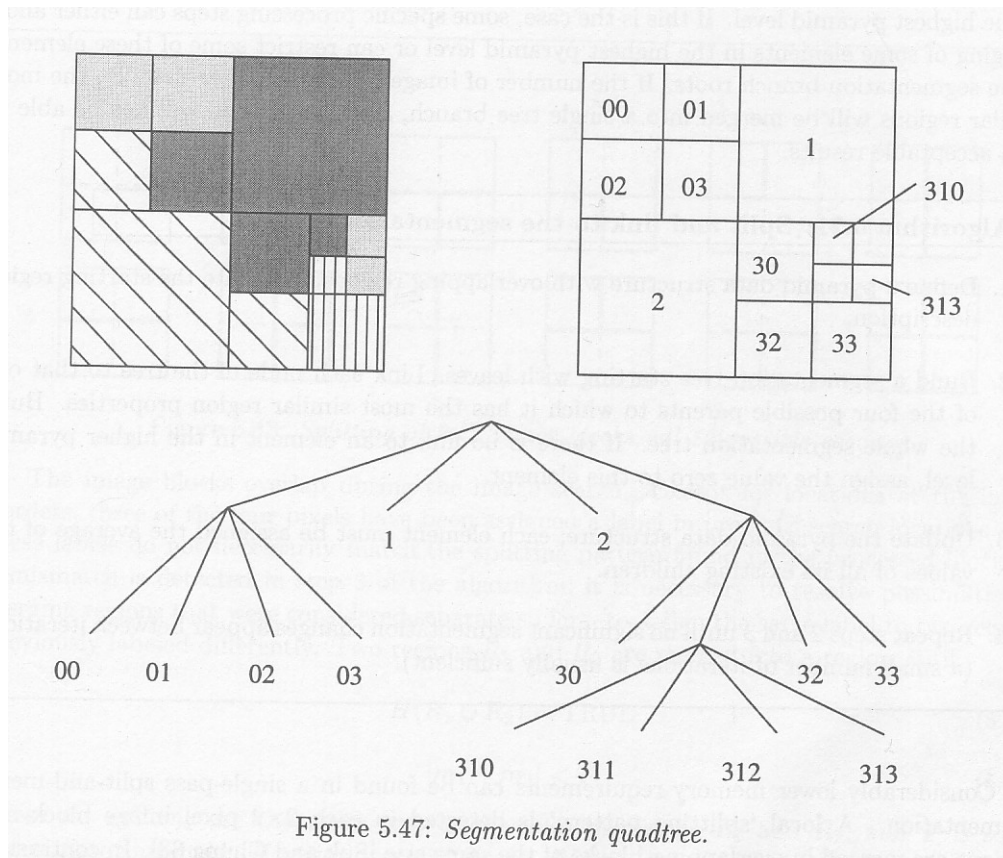


Figure 7.38 Example of split-and-merge algorithm. (From Fu, Gonzalez, and Lee [1987].)

Porovnávanie so vzorom (matching)

- Porovnávanie so vzorom možno použiť na lokalizáciu objektov so známym vzhľadom v obraze, na hľadanie špecifických vzorov, apod. Najlepšia zhoda je založená na nejakom kritériu optimality, ktoré záleží na vlastnostiach objektu a vzťahoch objektu.
- Algoritmus spočíva vo vyhodnotení kritéria zhody pre každú polohu a otočenie vzoru v obraze. Tie vyhodnotenie, ktoré presahujú stanovený prah, reprezentujú umiestnenie vzoru v obraze.



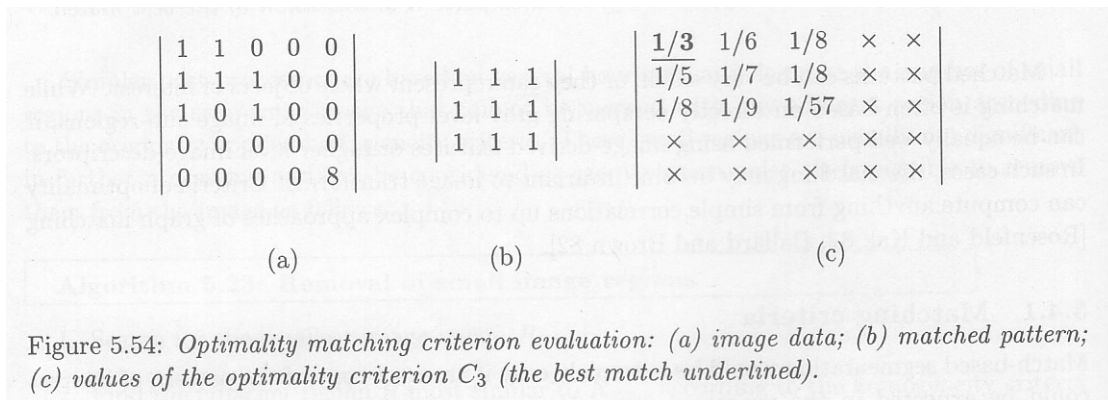
Figure 5.53: Segmentation by matching; matched pattern and location of the best match.

- Kritériá zhody možno definovať viacerými spôsobmi; špeciálne, korelácia medzi vzorom a skúmanými obrazovými dátami sa často používa ako všeobecné kritérium.

$$C_1(u, v) = \frac{1}{\max_{(i,j) \in V} |f(i+u, j+v) - h(i, j)| + 1}$$

$$C_2(u, v) = \frac{1}{(\sum_{(i,j) \in V} |f(i+u, j+v) - h(i, j)|) + 1}$$

$$C_3(u, v) = \frac{1}{(\sum_{(i,j) \in V} [f(i+u, j+v) - h(i, j)]^2) + 1}$$



- Praktické stratégie pre použitie matchingu:
 - o ak máme urobiť porovnanie vo všetkých polohách, je to výpočtovo veľmi náročné, aj keby sme už nerobili ďalšie geometrické transformácie ako otočenie alebo zväčšovanie či zmenšovanie vzoru. Preto sa používajú postupy, že ak sa dosiahne nezhoda, porovnanie sa preruší, alebo sa začne porovnávať na nižšej úrovni rozlíšenia (vychádza sa z toho, že zhoda sa mení postupne aj v okolí najväčšej zhody).